REMARKS

Claims 1-8, and 10-20 are pending in the application. Claims 1-8, and 10-20 stand rejected. Claim 9 has been canceled because it was inadvertently included as a duplicate of Claim 7. Applicant requests reconsideration of the application in view of the following remarks.

Double Patenting

The Examiner rejected Claim 9 for potential double patenting over Claim 7. Applicant has canceled Claim 9. Applicant therefore respectfully requests that the Examiner's double patenting rejection be withdrawn as moot.

Claim Rejections - 35 U.S.C. §101

The Examiner has rejected Claims 1-20 as being directed to non-statutory subject matter. Applicant respectfully traverses the rejection.

The Examiner bases his rejection on his assertion that the method of claims 1-19 and the system of Claim 20 do not actually construct a computer program. Applicant respectfully directs the Examiner's attention to the last clause of Claim 1, which states the step of "recursively selecting an implementation for each subtrait associated with at least one of the traits or the implementations *in order to construct a trait hierarchy that forms a computer program for solving the problem.*" (emphasis added). Claim 1 states that the method's step of recursively selecting an implementation for each subtrait constructs a trait hierarchy that forms a computer program for solving the problem. Thus, the method of claim 1 does actually form a computer program. Because Claims 2-8 and 10-19 depend from claim 1, the methods of Claims 2-8 and 10-19 also actually form a computer program.

Similarly, the system of Claim 20 comprises a "means for recursively selecting an implementation for each subtrait associated with at least one of the traits or the implementations *in order to construct a trait hierarchy that forms a computer program for solving the problem.*" (emphasis added). Thus, the system of claim 20 does actually form a computer program.

Because claims 1-20 achieve a tangible result by forming a computer program, Applicant respectfully requests withdrawal of the Section 101 rejection.

Claim Rejections - 35 U.S.C. §112

Claims 1-20 stand rejected under 35 U.S.C. § 112 for failing to particularly point out and distinctly claim the subject matter which the applicant regards as his invention. Applicant respectfully traverses the rejection. Applicant's invention is directed to a method of and system for constructing a computer program. The claims set forth method steps, and means in a system, specifying particular aspects of computer programming tasks which, taken together, improve upon prior art methods and systems for constructing computer programs.

As a preliminary matter, applicant believes that the Examiner has confused the breadth of the claims with indefiniteness. If the scope of the subject matter embraced by the claims is clear, and if applicant has not otherwise indicated that he intends the invention to be of a scope different from that defined in the claims, then the claims comply with 35 U.S.C. § 112. *See* M.P.E.P 2173.04.

In the instant case, the subject matter defined by the claims is clear and unambiguous, even though the claims are broad. Specifically with respect to claim 1, and by way of explanation and not intended to limit the claims, the method of constructing a computer program defines a set of traits wherein each trait characterizes a portion of a solution algorithm. Such traits can be, by way of example only, methods of a class in an object oriented programming environment. *See* the specification, *p*, 27 *ll*.6-8. The method of claim 1 then defines an interface for at least one trait, and implements at least one of the interfaces. *See* the specification, *p*, 28, *ll*. 1-2. The method then specifies a subtrait of one of the traits or the implementations, and selects a top-level trait that characterizes a solution to the problem. The subtrait can be, by way of example only, variables of the trait method or members of the trait. The method then selects an implementation for the top-level and for each subtrait of the top level trait. The method then recursively selects implementations for each subtrait of at least one of the traits or implementations already selected, depending on what subtraits exist for the particular traits and/or implementations already selected, until the necessary implementations are selected to create a computer program. Dependent claims further refine the method. For example, claim 13 adds an evaluation module to evaluate the constructed program, and an optimization technique to generate multiple computer programs to solve the same problem based on feedback from the evaluation module. Claim 15 permits interactive choices of trait implementations by users.

"The test for definiteness under 35 U.S.C. 112, second paragraph, is whether 'those skilled in the art would understand what is claimed when the claim is read in light of the specification.'" MPEP § 2173.02 (quoting *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1576, 1 USPQ2d 1081, 1088 (Fed. Cir. 1986)). *See also Metabolite Labs., Inc. v. Lab. Corp. of Am. Holdings*, 370 F.3d 1354, 1366, 71 USPQ2d 1081, 1089 (Fed. Cir. 2004) ("The requirement to 'distinctly' claim means that the claim must have a meaning discernible to one of ordinary skill in the art when construed according to correct principles. Only when a claim remains insolubly ambiguous without a discernible meaning after all reasonable attempts at construction must a court declare it indefinite.") (*cited with approval at* MPEP § 2173.02).

Applicant submits that the terms used in claims 1-20 particularly set forth the metes and bounds of the claimed invention, as required by § 112, and that a person having ordinary skill in the art of computer programming would be able to determine the scope and meaning of each claim taken as a whole. Applicant therefore respectfully submits that the claims meet the requirements of § 112, and respectfully requests withdrawal of the § 112 rejection.

Claim Rejections - 35 U.S.C. §103 (a)

Claims 1-20 stand rejected as obvious in view of U.S. Patent No. 7,047,169 ("Pelikan") in combination with U.S. Patent No. 6,263,325 ("Yoshida"). Applicant respectfully traverses the rejection.

Claim 1 recites a method of constructing at least one computer program comprising the step of "defining a programming interface for at least one of the traits." Neither Pelikan or Yoshida disclose, teach, or suggest, alone or in combination, this claimed feature.

Applicant agrees with the Examiner that Pelikan does not disclose this claimed feature. *See* Office Action at 6. However, the Examiner points to a user terminal disclosed in Yoshida as the equivalent of the "programming interface" of the claimed invention. However, a user terminal is not an "interface" as that term is used in claim 1 or the specification of the present application. The term "interface" is a term of art in object-oriented programming, used to refer to an abstract definition of, for example, a class of objects and methods that is not implemented within the interface. By way of example, the Sun Microsystems Java Glossary defines "interface" as "a Java keyword used to define a collection of method definitions and constant values. It can later be implemented by classes that define this interface with the 'implements' keyword." Sun Developers Network Glossary.

http://java.sun.com/docs/glossary.html. Neither Pelikan nor Yoshida teach or suggest "defining a programming interface" as the term "interface" is known to persons of ordinary skill in the art and as used in the specification. For this reason alone the Examiner's rejection is improper and should be withdrawn.

Similarly, neither Pelikan nor Yoshida teach or suggest "providing an implementation for at least one of the defined programming interfaces," because a programming interface is not disclosed. Because neither Pelikan nor Yoshida teaches nor suggests this programming element, the invention of claim 1 is not obvious in view of the combination of Pelikan and Yoshida.

Additionally and separately, Pelikan and Yoshida do not disclose the step of "recursively selecting an implementation for each subtrait associated with at least one of the traits or the implementations in order to construct a trait hierarchy that forms a computer program for solving the problem" of claim 1. The Examiner points to col. 3, lines 52-54 for such a recursively selecting step. However, no such step is disclosed at this section of Pelikan. In fact, that section makes reference to an "iterative process," which is known to persons of skill in the art to be contrasted with a recursive process. *See, e.g.,* Geek.com Glossary, http://www.geek.com/glossary.[1]

Pelikan discloses an iterative process, "generating a first set of solutions, selecting a second set from the first, fitting the second set with a probabilistic model that provides for 'chunking' whereby a plurality of variables may be merged into a single variable, using the model to generate a third set of solutions, and replacing at least a portion of the first set with the third set." Pelikan, *col.* 3, *ll.* 58-63. This is distinct from the recursive process of claim 1, which recurses through each subtrait associated with a trait or an implementation of the trait to select an implementation of the subtrait. As a recursive process, the method step has a natural end when

---

[1] Definitions from http://www.geek.com/glossary

"Iterative - This is a programming term. It refers to a process that can be described by a fixed number of variables and a set of rules that describes what happens to those variables to achieve the next step of the process. If the process is interrupted, it can be continued if the state of all the variables is known. Contrast this to a recursive process. An iteration is a single step."

"Recursive - A programming term describing a process that defers its operations as it runs, growing in memory size. Once all operations are deferred, there is a computation process that runs through the operations and figures out the values. If a recursive process is interrupted, there is not any easy way to recover it, as the operation is constructed on the fly with no fixed amount of variables. Contrast this to an iterative process."

there are no remaining subtraits for which implementations need be selected, based on the previously selected implementations.

This is different from an iterative process, which executes a number of times, independent of the results of prior executions. The iterative process of Peliken uses a fixed number of variables (the single set of solutions that is present in each iteration) and a set of rules that describes what happens to those variables to achieve the next step of the process (using the model to transform one set of solutions into the next set of solutions). In contrast, the recursive process defined in the claims defers its operations, i.e., its selection of subtrait implementations, as it runs, and then determines the values, i.e., (chooses the subtrait implementations. Because Peliken and Yoshida fail to teach or suggest this step of claim 1, claim 1 is not obvious in view of Peliken combined with Yoshida. Applicant respectfully requests that the rejection of claim 1 be withdrawn, and that claim 1 be allowed.

Claim 20 discloses a system for constructing at least one computer program comprising "an implementation for at least one of the defined programming interfaces", "a subtrait associated with at least one of the traits or the implementations," and "means for recursively selecting an implementation for each subtrait associated with at least one of the traits or the implementations in order to construct a trait hierarchy that forms a computer program for solving the problem." Similar to the analysis above for claim 1, Peliken and Yoshida do not disclose these elements of claim 20. Applicant respectfully requests that the rejection of claim 20 be withdrawn, and that claim 20 be allowed.

In addition, claims 1 and 20 define "a set of traits," in which a trait is defined in the specification as characterizing "a portion of a solution algorithm." This feature provides components that will be assembled (through later steps) into one or more solutions to the target problem, which may be any type of problem that has an algorithmic solution. In contrast, the Examiner relies on Peliken to teach a "first set of solutions" or "solution set for a problem." This is a set of **complete** candidate solutions to the target problem, where a "problem" is defined by Peliken as being "a problem defined over discrete variables" and not a "solution algorithm" as defined by the claims. Thus, Peliken fails to disclose this claimed element as alleged by the Examiner.

Claims 1 and 20 also define "selecting an implementation for each subtrait." In contrast, the Examiner relies on Peliken's "second set from the first" disclosure; however, this

11

element means that a subset of the "first set of solutions" can be selected, each of which is a complete candidate solution for the target problem. Merely, selecting a first set of solutions, is not the equivalent of selecting an implementation for each subtrait since "selecting an implementation for each subtrait" requires choosing one of the available implementations for each subtrait that will become part of a complete solution to the problem. Thus, Peliken fails to disclose this claimed element as alleged by the Examiner.
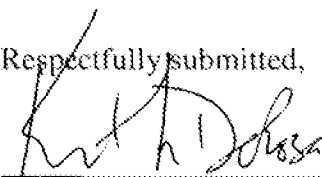
Because claims 2-8 and 10-19 depend from claim 1, claims 2-8 and 10-19 are allowable based at least upon their dependency upon an allowable independent claim. Applicant respectfully requests that the rejection of claims 2-8 and 10-19 be withdrawn, and that claims 2-8 and 10-19 be allowed.

The remaining references cited (but not applied) have been reviewed. These references are not deemed to be material to the patentability of the claimed invention. For the foregoing reasons, the claims as they now stand are patentable over the art of record, and withdrawal of the rejections and allowance of all pending claims is earnestly solicited.

****

It is respectfully submitted that the claims in the application are allowable.

Reconsideration and withdrawal of all rejections are respectfully requested. Favorable notice to

this effect and early Notice of Allowance are earnestly solicited.

Should the examiner have any questions and in order to expedite prosecution of

this Application, the Examiner is encouraged to contact the undersigned directly.

Respectfully submitted,

Kenneth R. DeRosa, Reg. No. 39,549
Stuart D. Rudoler, Reg. No. 45,059
Attorney for Applicant
Rudoler & DeRosa LLC
2 Bala Plaza, Suite 300
Bala Cynwyd, PA 19004
Tel: 610-660-7753; Fax: 267-200-0796